UnityBasic — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-07-03 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P034 / 근거 수준: C / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 핵심 메카닉스

대사 진행 (시트 → 줄 → 화면)

- 의도(Why): 시나리오를 시트에서 편집하고 Unity는 표시만 담당하게 해 VN 제작의 반복 수정 비용을 줄인다. [추정: 외부 시트 채택]
- 설계(How): 시작 시 코루틴이 게시 CSV를 `UnityWebRequest`로 내려받아 `id → DialogueLine` 사전에 저장하고 `startId`(PJ_03 값 10001) 줄을 표시한다. `Next()`는 현재 줄의 `nextId`가 비어 있거나 "0/end/none/-"이면 종료 이벤트를, 아니면 다음 id를 표시한다. `Jump(id)`는 분기용 직접 이동이다. 줄 필드: id, nextId, speaker, text, portrait, background, character. 선택지 대신 단순 연결 리스트 구조를 택한 이유는 기초 실습 범위 한정으로 추정한다. [근거: DialoguePlayer.cs; PJ_03.unity] [추정: 범위 한정]
- 필요(What): 구글시트 게시 CSV 주소, 씬당 `DialoguePlayer` 1개.
- 작업 방법: 시트 작성 → `DialogueManager` 오브젝트에 컴포넌트 부착 → URL/startId 입력 → 실행.
- 완료 기준: 로드 로그 "[대사] N줄 로드 완료" 후 첫 줄 표시, 마지막 줄에서 `onEnd` 호출.
- 리스크/대안: 로드 실패 시 에러 로그만 남고 진행 불가 → 로컬 캐시/TextAsset 대안. 없는 id는 에러 로그 후 현재 줄 유지.
- 근거: [근거: DialoguePlayer.cs]

CSV 파서

- 의도(Why): 시트 셀 안의 쉼표/줄바꿈(대사 문장)이 깨지지 않게 한다.
- 설계(How): 문자 단위 상태 파서. 따옴표 안에서는 쉼표/개행을 필드 문자로 취급한다. 열은 헤더 이름으로 찾으며(id/대사id, next/다음/다음대사id, speaker/화자, text/대사/내용, portrait/포트레이트, background/배경, character/캐릭터), 열 순서와 한영 표기에 무관하다. id가 빈 행은 건너뛴다.
- 필요(What): 헤더 행이 있는 CSV.
- 작업 방법: 시트 헤더를 위 이름 중 하나로 작성.
- 완료 기준: 열 순서를 바꿔도 동일하게 동작.
- 리스크/대안: 따옴표 이스케이프("")는 처리하지 않아 따옴표 포함 대사가 깨질 수 있음. 이중 따옴표 처리 추가 대안.
- 근거: [근거: DialoguePlayer.cs ParseCsv]

화면 렌더링

- 의도(Why): 로직과 UI를 분리해 UI 교체에도 로직을 재사용한다.

- 설계(How): `DialogueView`가 `Awake`에서 `onLine`에 `Render`를 등록하고 다음 버튼 클릭을 `Next`에 연결한다. 화자/본문은 TMP_Text, 이미지 3종(초상, 배경, 캐릭터)은 키가 비면 숨기고, 있으면 `Resources.Load<Sprite>(폴더/키)`로 로드한다. 폴더 필드를 비우면 Resources 루트에서 찾는다. 실패 시 경고 후 숨김.
- 필요(What): 인스펙터 참조 연결, 스프라이트 임포트 타입 Sprite.
- 작업 방법: Canvas에 BG, Char, Portrait, 대사 TMP, 이름 TMP, 버튼 배치 후 드래그 연결.
- 완료 기준: 줄마다 텍스트/이미지 갱신, 빈 키는 이미지 숨김.
- 리스크/대안: Resources 전체 로드는 규모 확대 시 비효율 → Addressables.
- 근거: [근거: DialogueView.cs; PJ_03.unity]

주기 스폰

- 의도(Why): Update, Time.deltaTime, Instantiate 기초를 익힌다. [추정: 학습 코드 형태]
- 설계(How): `timer`에 deltaTime 누적, `spawnCooldown`(PJ_01 값 2초) 이상이면 0으로 리셋하고 `spawnPoints`의 위치/회전으로 `prefabToSpawn`을 Instantiate. 생성 제한/파괴 로직 없음.
- 필요(What): 프리팹(`dudonban 1`), 스폰 지점 Transform.
- 작업 방법: 빈 오브젝트에 스크립트 부착 후 프리팹/지점 지정.
- 완료 기준: 2초마다 오브젝트 생성.
- 리스크/대안: 무한 생성으로 성능 저하 → 수명/최대 수 제한.
- 근거: [근거: SpawnManager.cs; PJ_01.unity]

2. 구현 방법

- 구조: 씬별 독립. 공용 상태/저장 없음. [근거: Scripts 3개뿐]
- 상태 머신: `Current` 줄이 상태. null이면 대기/종료. [근거: DialoguePlayer.cs]
- 시스템 연결: UnityEvent(`onLine`, `onEnd`)와 버튼 `onClick`. [근거: DialoguePlayer.cs]
- 데이터 파이프라인: 구글시트 → 게시 CSV → 런타임 다운로드. [근거: PJ_03.unity]
- 도구: Unity 6000.3.19f1, TextMeshPro, NotoSansKR SDF. [근거: raw/codex1.md; Assets/Fonts]

3. 시스템 간 상호작용과 밸런스 기준

- 시스템 간 상호작용은 대사↔UI뿐이다. 밸런스 변수는 없다.
- 유일한 수치: `spawnCooldown` 2초, 시작 id 10001. [근거: PJ_01.unity; PJ_03.unity]

4. 게임의 나머지 부분의 방향

- 콘텐츠 구조: 실습 씬 모음. 스토리/챕터 없음. [추정]
- 메타 진행, 저장, 설정: 기록 없음(구현 없음).
- UI/UX: 클릭 진행형 VN 패널. 자동 진행, 로그, 스킵, 선택지는 미구현. [근거: DialogueView.cs]
- 아트/사운드: 정지 이미지 3계층, 사운드 없음.
- 향후 방향(제안): 선택지 UI, 오프라인 폴백 [제안값] 없음, 필요 시 다음 VN 프로젝트로 이식. [추정]

5. 제작 과정 설계

- 순서: 씬 실습(PJ_00→01→02) → 스크립트(스폰) → UI/VN(PJ_03) → 백업 커밋. [추정: 씬 번호/커밋 시점]
- 병렬 가능: 시트 대사 작성과 이미지 준비는 코드와 병렬.
- AI/도구 분담: 기록 없음. 코드 주석의 설명 투가 AI 또는 강의 예제와 유사하나 단정하지 않는다. [추정]
- 검증: 에디터 실행 후 콘솔 로그 확인(로드 줄 수 로그가 검증 수단). [근거: DialoguePlayer.cs]
- 인수인계: README 한 줄, HANDOFF 없음.

6. 세부기획서 목록

문서	정할 것
03_세부기획서_대사시스템데이터	시트 열 명세, 리소스 키 규칙, 씬 구성, 검수 기준